REMARKS

The claims are claims 1 to 5, 7, 13 and 19 to 24.

The application has been further amended to correct minor errors. Paragraph [01] is amended to delete the Attorney Docket numbers of the referenced priority applications. Paragraph [26] is amended to update the status of the cited co-pending application and to delete the Attorney Docket number. Paragraph [75] is amended to correct a minor typographic error. Paragraph [85] is amended to correct a Figure reference. Element 850 is illustrated in Figure 27 and not in Figure 30.

Claims 1 to 5, 7, 19 and 20 have been amended. Claims 6, 8 to 12 and 14 to 18 are canceled. New claims 21 to 24 are added. Claim 1 is amended to include limitations previously recited in canceled claim 6. Claims 2 to 5 and 7 are amended to change the preamble recitation of "processing engine" to "digital system" as recited in base claim 1. New claims 21 to 24 recite subject matter not previously claimed.

Paragraph 13 of page 4 of the OFFICE ACTION stated that claim 6 would be allowable if rewritten into independent form including all limitations of its base claim. As amended claim 1 has this scope. Accordingly, claim 1 is allowable.

Claims 2 to 5, 7, 13, 19 and 20 are allowable by dependence upon allowable claim 1.

New claims 21 to 24 recite subject matter not made obvious by the references cited in the OFFICE ACTION. Claims 21 and 23 recite a fixed correspondence between the physical data register and the logical register and permitting the instruction to access the physical data registers corresponding to the logical registers when no conflict is detected. This differs from both register renaming methods disclosed in Smith et al. Smith et al states at the first column of page 1615, third paragraph:

> "When an instruction is decoded, its logical result register (destination) is assigned a physical register from a *free list*, i.e., the physical registers that currently do not have a logical value assigned to them, and the mapping table is adjusted to reflect the new logical to physical mapping for that register. In the process, the physical register is removed from the free list. Also, as part of the rename operation, the instruction's source register designators are used to look up their current physical register names in the mapping table. These are the locations from which the source operand values will be read. Instruction dispatch is temporarily halted if the free list is empty."

This disclosure of Smith et al teaches a changing correspondence between physical and logical registers contrary to the recitations of claims 21 and 23. This disclosure of Smith et al also teaches employing the changeable mapping between physical and logical registers even in the absence of a register access conflict. Smith et al states at the last full paragraph on the second column of page 1615 and the paragraph bridging pages 1615 and 1616:

> "The second method of renaming uses a physical register file that is the same size as the logical register filed, and the customary one-to-one relationship is maintained. In addition, there is a buffer with one entry per active instruction (i.e., an instruction that been dispatched for execution but which has not yet committed.) This buffer is typically referred to as a *reorder buffer* [32], [55], [57] because it is also used to maintain proper instruction ordering for precise interrupts.
>
> "Fig. 6 illustrates a reorder buffer. It is easiest to think of it as a FIFO storage, implemented in hardware as it circular buffer with head and tail pointers. As instructions are dispatched according to sequential program order, they are assigned entries at the tail of the reorder buffer. As instructions complete execution, their result values are inserted into their previously assigned entry, whatever it may happen to be in the reorder buffer. At the time an instruction reaches the head of the reorder buffer, if it has completed execution, its entry is removed from the buffer and its result value is placed in the register file. An

> incomplete instruction blocks the head of the reorder buffer until its value arrives."

This second method disclosed in Smith et al thus always uses the reorder buffer even when no conflicts are detected. Accordingly, claims 21 and 23 are allowable over Smith et al.

Claims 21 and 23 recite further subject matter not made obvious by Smith et al. Claims 21 and 23 recite the stalls upon RAW hazards and the length of time the result is stored in the shadow register upon WAR and WAW hazards is specified by the stall vector. Since the disclosed instruction pipeline may read and write to address registers during the P3 pipeline stage (see application Table 1 on page 12) the register access conflict may be resolved before the second instruction completes. However, the second method of Smith et al which maintains "the customary one-to-one relationship" between physical and logical registers does not store the result from the reorder buffer until the instruction is complete. Thus this invention may operate with fewer shadow registers than the depth of the reorder buffer of Smith et al because instructions may not be held as long. Thus claims 21 and 23 are allowable over Smith et al.

Claims 22 and 24 recite subject matter not made obvious by Smith et al. Claims 22 and 24 recite that the number of shadow registers is less than the maximum number of instructions "that can simultaneously cause register access conflicts." On the contrary, Smith et al teaches "a buffer with one entry per active instruction (i.e., an instruction that been dispatched for execution but which has not yet committed.)" In accordance with the recitations of claims 22 and 24 enough shadow registers may be provided for most situations that generate conflicts without requiring enough shadow registers for all possible cases as taught in Smith et al. Thus claims 22 and 24 are allowable over Smith et al.

Claims 22 and 24 recite further subject matter not made obvious by Smith et al. Claims 22 and 24 recite stalling the instruction for the number of pipeline stages of the stall vector if no shadow register is free. Smith et al includes no teaching of any action to be taken if the reorder buffer is full. Thus claims 22 and 24 are allowable over Smith et al.

The Applicant respectfully submits that all the present claims are allowable for the reasons set forth above. Therefore early reconsideration and advance to issue are respectfully requested.

If the Examiner has any questions or other correspondence regarding this application, Applicant requests that the Examiner contact Applicant's attorney at the below listed telephone number and address to facilitate prosecution.

Respectfully submitted,

Robert D. Marshall, Jr.

Robert D. Marshall, Jr.
Reg. No. 28,527

Texas Instruments Incorporated
P.O. Box 655474 M/S 3999
Dallas, Texas 75265
(972) 917-5290
Fax: (972) 917-4418